Virtual Data Integration

Helena Galhardas | Paulo Carreira
Agenda

- Terminology
- Conjunctive queries and Datalog
- Virtual Data Integration Architecture
Building a Data Integration System

Create a middleware mediator or data integration system over the sources

- Can be warehoused (a data warehouse) or virtual
- Presents a uniform query interface and schema
- Abstracts away multitude of sources; consults them for relevant data
  - Unifies different source data formats (and possibly schemas)
  - Sources are generally autonomous, not designed to be integrated
- Sources may be local DBs or remote web sources/services
- Sources may require certain input to return output (e.g., web forms)
  - binding patterns describe these
Logical components of a virtual integration system

**Source descriptions**
Specify the properties of the sources the system needs to know to use their data. Main component is **semantic mappings** that specify how attributes in the sources correspond to attributes in the mediated schema, how to resolve differences in how data values are specified in different sources.

Other info is whether sources are complete.

**Mediated Schema**
Is built for the data integration application and contains only the aspects of the domain relevant to the application. Most probably will contain a subset of the attributes seen in sources.

**Wrappers**
Programs whose role is to send queries to a data source, receive answers and possibly apply some basic transformation to the answer.
Example: a data integration scenario

Movies: Title, director, year, genre
Actors: title, actor
Plays: movie, location, startTime
Reviews: title, rating, description

S1: Movies: name, actors, director, genre
S2: Cinemas: place, movie, start
S3: Cinemas in NYC: cinema, title, startTime
S4: Cinemas in SF: location, movie, startingTime
S5: Reviews: title, date grade, review
Semantic mappings

Semantic mappings in the source descriptions describe:

- the **relationship** between the sources and the mediated schema, for example:
  - mapping of source S1 states it contains movies; the attribute name in Movies maps to attribute title in the mediator Movie relation; the Actors relation in the mediated schema is a projection of the Movies source on the attributes name and actors

- Whether the sources are **complete**, for example:
  - Source S2 may not contain all the movie showing times in the entire country
Components of a data integration system

Query (formulated on mediated schema)

Query reformulation

Logical query plan

Query optimizer

Physical query plan

Execution engine

Replanning request

wrapper

source

wrapper

source

wrapper

source

wrapper

source
Query reformulation (1)

- Rewrite the user query that was posed in terms of the relations in the mediated schema, into queries referring to the schemas of data sources.

- Result is called a **logical query plan**:  
  - Set of queries that refer to the schemata of the data sources and whose combination will yield the answer to the original query.
Query reformulation (2)

- **Ex:** `SELECT title, startTime
  FROM Movie, Plays
  WHERE Movie.title = Plays.movie
  AND location = 'New York'
  AND director = 'Woody Allen'

- Tuples for Movie can be obtained from source S1 but attribute title needs to be reformulated to name.

- Tuples for Plays can be obtained from S2 or S3. Since S3 is complete for showings in NY, we choose it.

- Since source S3 requires the title of a movie as input and the title is not specified in the query, the query plan must first access S1 and then feed the movie titles returned from S1 as inputs to S3.
Query optimization

- Accepts a logical query plan as input and produces a physical query plan
  - Specifies the exact order in which sources are accessed, when results are combined, which algorithms are used for performing operations on the data

- Ex:
  - The optimizer will decide the join algorithm to combine results from S1 and S3. It may stream tuples from S1 and input them into S3, or it may batch them up before sending them to S3
Query execution

- Responsible for the execution of the physical query plan
  - Dispatches the queries to the individual sources through the wrappers and combines the results as specified by the query plan.

- Also may ask the optimizer to reconsider its plan based on its monitoring of the plan’s progress (e.g., if source S3 is slow)